# Information-Preserving Nonlinear Transport via Adaptive Walsh Sign Geometry


## Abstract

Conventional neural networks derive nonlinear expressivity largely through information-destroying activations such as ReLU, sigmoid saturation, or max pooling. These operations partition or collapse regions of representation space. While effective, they introduce irreversible information loss and progressively degrade invertibility across depth.

This note explores an alternative viewpoint: nonlinear computation through *information-preserving geometric transport*. In this framework, nonlinear behavior arises not from thresholding or saturation, but from input-dependent modulation of orthogonal transport operators.

A particularly simple and interesting form is:

$$y = HD(s(x))\, x$$

where:

- H is a Walsh-Hadamard transform,
- D(s(x)) is a diagonal ±1 matrix derived from the input,
- s(x) extracts geometry-dependent sign structure from the representation.

This construction creates nonlinear adaptive transport while approximately preserving norm and information.


---

# 1. Motivation

A deep stack of purely linear layers collapses to a single affine map:

$A_n A_{n-1} \cdots A_1 = A_{eff}$

Thus deep linear systems cannot accumulate nonlinear geometric structure.

Standard neural networks avoid this collapse using nonlinear activations such as:

$\mathrm{ReLU}(x) = \max(0,x)$

However, these functions are many-to-one mappings:

- negative regions collapse,
- saturation destroys magnitude information,
- interpolation structure is disrupted.

This raises a natural question:

> Can nonlinear expressivity be achieved through geometric warping without information destruction?

The key realization is:

> Affine collapse is prevented by nonlinearity itself, not by information loss.

Any transformation violating superposition:

$f(x+y) \neq f(x)+f(y)$

escapes affine reducibility even if it remains invertible or approximately norm-preserving.

---

# 2. Nonlinearity Through Adaptive Transport

Instead of local scalar activations, consider input-dependent transport operators.

For fixed diagonal D,

HDH

is a linear orthogonal transform.

But if D depends on the input:

y = HD(s(x))Hx

the overall mapping becomes nonlinear because the transport operator itself varies with the representation.

For nearby inputs:

- nearby operators may be selected,
- geometry changes smoothly,
- nonlinear curvature accumulates gradually.

The network computes through:

- adaptive coordinate transport,
  rather than:
- threshold partitioning.

---

# 3. Why the Placement of D Matters

An important observation is that:

DH

and

HD

behave very differently.

Since D is diagonal:

- DH flips rows of H,
- HD flips columns of H.

This distinction is geometrically significant.

In:

BD(x)HA

the modulation acts primarily on output coordinates.

But in:

BHD(x)HA

the modulation occurs inside the transport basis itself.

The operator:

HD(x)H

therefore behaves like:

- adaptive spectral modulation,
- global coordinate warping,
- distributed reflection geometry.

This is substantially richer than direct output gating.

---

# 4. Geometry of HD(s(x))H

The Walsh-Hadamard transform mixes all coordinates globally.

A single sign flip in the Walsh basis therefore influences all output dimensions after remixing.

Consequently:

- modulation effects become distributed,
- nearby transport operators remain correlated,
- nonlinear behavior emerges through changing interference structure.

Importantly, for fixed D:

$$(HDH)^T(HDH)=I$$

so the transport remains orthogonal.

Thus:

- norm is preserved,
- dimensional collapse is avoided,
- nonlinear behavior arises entirely from adaptive geometry.

This differs fundamentally from ReLU-style computation.

---

# 5. Smooth vs Quantized Geometry

An early idea considered selecting D from a large discrete pool of random sign matrices.

However this produced undesirable behavior:

- nearby inputs could trigger unrelated transport operators,
- interpolation became discontinuous,
- small input changes caused global scrambling.

The important refinement was:

> Nearby inputs should produce nearby transport geometries.

Thus the sign structure should depend coherently on representation geometry itself.

This shifted the architecture from:

- combinatorial transport branching,
  to:
- smoothly evolving adaptive transport.

---

# 6. Candidate Definitions of s(x)

Several geometrically meaningful options emerge.

---

## A. Coordinate Sign Geometry

$$s(x) = \text{sign}(x)$$

giving

$$D(s(x)) = \text{diag}(\text{sign}(x_i))$$

This is the simplest form.

Advantages:

- cheap,
- local,
- correlated for nearby inputs,
- naturally orientation-sensitive.

The transport operator adapts directly to the sign structure of the representation.

---

## B. Spectral Sign Geometry

$$s(x) = \text{sign}(Hx)$$

Now the signs are extracted from globally mixed coordinates.

Each sign already depends on all input dimensions.

This creates:

- globally coherent transport adaptation,
- distributed geometric sensitivity,
- smoother operator evolution.

This version may be particularly attractive because the adaptive geometry is itself defined in the Walsh basis.

---

## C. Mixed Projection Geometry

$$s(x) = \operatorname{sign}(RHx)$$

where R is a fixed random or learned orthogonal projection.

This produces:

- distributed orientation sensing,
- stable correlated sign patterns,
- richer adaptive transport geometry.

The resulting operator depends on large-scale representation structure rather than raw coordinates.

---

# 7. Interpretation as Adaptive Spectral Transport

The transform

$$HD(s(x))H$$

can be interpreted as:

- an input-conditioned spectral transport operator,
- a distributed reflection geometry,
- or an adaptive wave-scattering transform.

The computation proceeds by:

1. globally mixing coordinates,
2. extracting representation geometry,
3. modulating transport structure,
4. remixing globally.

Nonlinearity emerges from:

- changing coordinate geometry,
  not:
- information deletion.

---

# 8. Comparison with Standard Neural Networks

Standard deep networks compute primarily through:

- thresholding,
- sparsification,
- partitioning representation space into affine regions.

The present framework instead computes through:

- adaptive orthogonal transport,
- curvature accumulation,
- smooth coordinate warping.

The representation evolves through:

- changing geometry,
  rather than:
- progressive information collapse.

---

# 9. Open Questions

Several important questions remain:

### Stability

How rapidly should transport geometry vary with input?

### Continuity

Can discontinuities near sign boundaries be softened?

### Learnability

Can useful transport geometries emerge through optimization?

### Expressivity

How does adaptive orthogonal transport compare to conventional activations in representation power?

### Geometry

What forms of curvature and manifold structure are naturally induced?

---

# 10. Conclusion

The transform

$$y = HD(s(x))Hx$$

illustrates a different route to nonlinear computation.

Rather than obtaining expressivity through irreversible activation collapse, nonlinear behavior emerges from:

- input-conditioned transport geometry,
- adaptive spectral modulation,
- distributed orthogonal warping.

This suggests a broader class of deep architectures based on:

- geometry-preserving nonlinear transport,
  rather than:

- threshold-driven feature destruction.

Such systems may provide an alternative foundation for deep representation learning based on smooth adaptive geometry instead of combinatorial partitioning.